Help Page

Auxiliary RepWords 1.0 Program

Auxiliary RepWords 1.0 program is intended for (a)computational speed tests of the generalized Ruzzo-Tompa algorithm and the divide-and-conquer algorithm; (b)computing the score threshold to be used with the RepWords 1.0 program; (c)computation of composition frequencies for an arbitrary sequence.

The program is focused on applying the generalized Ruzzo-Tompa algorithm to repeats finding problem.

Auxiliary RepWords 1.0 program can be downloaded from the URL:

http://www.ncbi.nlm.nih.gov/CBBresearch/Spouge/html_ncbi/html/index/software.html#18

Instructions for the installation can be found here.

Usage.

The program is run with parameters separated by spaces. Each parameter of the command line is the pair:

-<parameter name> <parameter value>

Parameters.

The program can be executed in the three different modes:

- mode A: the test mode. In this mode the program runs a speed test for comparison of the two algorithms: “Generalized Ruzzo-Tompa algorithm” and “Divide-and-conquer algorithm”.

- mode B: the score threshold calculation mode. In this mode the program calculates a repeat score threshold corresponded to an input coverage value. This score threshold can be used for a filtering of repeats by their scores. The score threshold is a parameter of the RepWords 1.0 program.
- mode C: the composition frequencies calculation mode. The program extracts composition frequencies from an input sequence in FASTA format. The result can be used as an input for mode B.

Which mode is running is determined by the parameter "-mode" that is a required parameter.

The opening gap penalty **d1** and the extending gap penalty **d2** (defined in Mode A and Mode B of the program) assume the following convention: a gap of length **k** is penalized as **d1+d2*k**.

Mode A (the test mode)

-mode <a mode of the program>

- Defines the mode and must have the value "test".
- The parameter is required.

-w <the word-length>

- Must be a positive integer number.
- The parameter is required.

-gap_open <the opening gap penalty>

- Must be a non-negative integer number.
- The parameter is required.

-gap_extend <the extending gap penalty>

- Must be a positive integer number.
- The parameter is required.

-scoring_matrix <a name of an input file with the scoring matrix>

- The format of the file as follows. The first line contains a positive integer number **B** of letters in the alphabet. The rest of the file is a **B**x**B** table with **B** rows and **B** columns. The element from the row **a** and the column **b** of the table is an integer number corresponded to the similarity score between the letters with the order numbers **a** and **b**.
- The parameter is required.

-frequencies_input <a name of an input file with the background frequencies>

- The format of the file as follows. The first line contains a positive integer number **B** of letters in the alphabet. The next **B** lines contain the background frequencies: one real number per each line. The sum of the background frequencies must be equal to 1.
- The parameter is required.

-sequence_length <length of test sequences>

- Must be a positive integer number.
- The parameter is required.

-sequences_number <a number of test sequences>

- Must be a positive integer number.
- The parameter is optional (the default value is 1).

-trials <a number of trials>

- Must be a positive integer number.
- The first trial uses the input sequence length (defined by the parameter "-sequence_length"). The sequence length of the **K**th trial is the input sequence length multiplied by **2^(K-1)**.
- For example, in the case "-sequence_length 128 -sequences_number 100 -trials 5", the program performs 5 different tests for the lengths 128, 256, 512, 1024, 2048 and each test generates 100 random sequences of the corresponding length (incremented by **w**) according to the background frequencies.
- The parameter is optional (the default value is 1).

-screen_output <screen output flag>

- Determines whether the program outputs the resulted maximal intervals (repeats) on the screen (the value "true") or not (the value "false").
- The parameter is optional (the default value is "false").

-srand <the randomization number>

- Must be a non-negative integer number.
- Defines a seed for pseudorandom numbers generated in the program.
- If the parameter equals 0, then the randomization number is generated inside the program (from the system time).
- The randomization number is outputted on the screen.
- The program exactly reproduces the output if the same randomization number and other parameters are used.
- The parameter is optional (the default value is 0).

Mode B (calculation of the score threshold)

-mode <a mode of the program>

- Defines the mode and must have the value "scorethreshold".
- The parameter is required.

-input_w <a name of a file with a list of word-lengths **w**>

- Format of "-input_w" file as follows. The file consists of lines and each line begins with a positive integer value of **w**. The rest of the line is ignored. The program reads all the lines and all **w** listed are used in the calculation.
- The parameter is required.

-gap_open <the opening gap penalty>

- Must be a non-negative integer number.
- The parameter is required.

-gap_extend <the extending gap penalty>

- Must be a positive integer number.
- The parameter is required.

-scoring_matrix <a name of an input file with the scoring matrix>

- The parameter has the same meaning as in Mode A.
- The parameter is required.

-frequencies_input <a name of an input file with the background frequencies>

- The parameter has the same meaning as in Mode A.
- The parameter is required.

-sequence_length <length of test sequences>

- Must be a positive integer number.
- The parameter is required.

-sequences_number <a number of test sequences>

- Must be a positive integer number.
- The parameter is required.

-coverage <input coverage>

- Must be a positive real number.
- The coverage is used for the score threshold calculation.
- The parameter is optional (no score threshold is calculated if the parameter is not defined).

-distribution_output <a name of a file with a resulted distribution of the coverage for different scores>

- The parameter is optional (no file output is generated if the parameter is not defined).

-gapped <gap penalties flag>

- Defines whether the repeats are gapped (the value "true") or not (the value "false").

- The parameter is optional (the default value is "true").

-srand <the randomization number>

- The parameter has the same meaning as in Mode A.
- The parameter is optional (the default value is 0).

Mode C (composition frequencies calculation mode)

-mode <a mode of the program>

- Defines the mode and must have the value "fastacomp".
- The parameter is required.

-alphabet_yes <a name of an input file with a list of permitted letters>

- The parameter is required.

-FASTA_input <a name of an input file with a sequence>

- The parameter is required.

-frequencies_output <a name of an output file with the resulted frequencies>

- The parameter is required.

Output.

The program outputs some information on the screen and into files depending on the parameters.

Screen output.

Screen output in mode A:

- The randomization number.
- Calculation progress.
- Calculation times for each method for different sequence lengths.
- Calculated maximal intervals in the case when the parameter "-screen_output" is "true".

Screen output in mode B:

- The randomization number.
- Calculation progress.
- The calculated score threshold.
- An exact coverage value corresponded to the calculated score threshold.

Screen output in mode C:

- A total number of allowed letters from an input sequence.
- Calculated frequencies of the letters.

File output.

File output in mode A.

No file output generated in mode A.

File output in mode B.

The program outputs a single file in mode B. The file contains

- Tail distribution of the coverage for different repeat scores. The distribution is calculated for an input set of **w** as it defined by the parameter "-input_w".
- The calculated score threshold.
- An exact coverage value corresponded to the calculated score threshold.

File output in the mode C.

The program outputs the calculated frequencies of letters into a file with the name defined by the parameter "-frequencies_output". The file has the same format as an input file with frequencies used with the program RepWords 1.0.

Examples of the command line.

Mode A

-mode test -w 16 -gap_open 5 -gap_extend 2 -scoring_matrix matr4.in -frequencies_input RR4.in -sequence_length 128 -sequences_number 100 -trials 5 -screen_output false -srand 76832338

The program performs tests for 5 different sequence lengths 128, 256, 512, 1024, 2048 and for each test the program generates 100 random sequences of the corresponding length (incremented by **w=16**) according to background frequencies defined in the file "RR4.in":

```
4
0.25
0.25
0.25
0.25
```

The program uses the randomization seed 76832338 and does not output resulted maximal intervals on the screen. The affine gap penalties are 5/2 (a gap of length **k** is penalized as **5+2*k**); the scoring matrix is extracted from the file "matr4.in":

```
4
4     -5    -5    -5
-5    4     -5    -5
-5    -5    4     -5
-5    -5    -5    4
```

The program outputs the calculation times for each sequence length and method on the screen.

Mode B

-mode scorethreshold -input_w w10.in -gap_open 5 -gap_extend 2 -scoring_matrix matr4.in -frequencies_input RR4.in -sequence_length 10000 -sequences_number 2 -coverage 0.05 -distribution_output coverages_test.out -gapped true -srand 76832338

The program calculates the coverage values for the set of **w** inputted from the file "w10.in":

```
1       w1.txt
2       w2.txt
3       w3.txt
4       w4.txt
5       w5.txt
6       w6.txt
7       w7.txt
8       w8.txt
9       w9.txt
10      w10.txt
```

The program generates 2 sequences (the parameter "-sequences_number 2") of length 10000 (incremented by **w**) for each **w** from the interval [1,10]. The program uses the randomization seed 76832338. The affine gap penalties are 5/2 (a gap of length **k** is penalized as **5+2*k**); the scoring matrix is extracted from the file "matr4.in" and the background frequencies - from the file "RR4.in" (please see examples of the input files above in the description for Mode A).

The program computes a score threshold for the coverage value 0.05 (the parameter "-coverage 0.05") and outputs the results into the file "pvalues_test.out".

Mode C

-mode fastacomp -alphabet_yes alphabet_ACGT.in -FASTA_input seq.in -frequencies_output RR4.out

The program inputs permitted alphabet letters from the file "alphabet_ACGT.in":

```
4
ACGT
```

and the input sequence - from the file "seq.in":

```
>gi|224589810|ref|NC_000019.9| Homo sapiens chromosome 19, GRCh37.p2 primary
reference assembly fragment
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNGATCACAGAGGCTGGGCTGCTCCCCACCCTCTGCACACCTCCTGCTTCTAACAGCAGAGC
TGCCAGGCCAGGCCCTCAGGCAAGGGCTCTGAAGTCAGGGTCACCTACTTGCCAGGGCCGATCTTGGTGC
CATCCAGGGGGCCTCTACAAGGATAATCTGACCTGCAGGGTCGAGGAGTTGACGGTGCTGAGTTCCCTGC
ACTCTCAGTAGGGACAGGCCCTATGCTGCCACCTGTACATGCTATCTGAAGGACAGCCTCCAGGGCACAC
AGAGGATGGTATTTACACATGCACACATGGCTACTGATGGGGCAAGCACTTCACAACCCCTCATGATCAC
GTGCAGCAGACAATGTGGCCTCTGCAGAGGGGGAACGGAGACCGGAGGCTGAGACTGGCAAGGCTGGACC
TGAGTGTCGTCACCTAAATTCAGACGGGGAACTGCCCCTGCACATAGTGAACGGCTCACTGAGCAAACCC
CGAGTCCCGACCACCGCCTCAGTGTGGTCTAGCTCCTCACCTGCTTCCATCCTCCCTGGTGCGGGGTGGG
CCCAGTGATATCAGCTGCCTGCTGTTCCCCAGATGTGCCAAGTGCATTCTTGTGTGCTTGCATCTCATGG
AACGCCATTTCCCCAGACATCCCTGTGGCTGGCTCCTGATGCCCGAGGCCCAAGTGTCTGAT
```

The composition frequencies are outputted into the file "RR4.out".

Files and Installation

The files in the download directory include:

1. auxiliary_repwords_1.0_WINDOWS.zip: Windows executable.
2. auxiliary_repwords_LINUX_1.0.zip: LINUX executable.
3. auxiliary_repwords_cpp_files.zip: C++ source files.
4. auxiliary_repwords_examples_files.zip: contains the following sample files:

    - w10.in: an example of an input file for the parameter "-input_w".
    - matr4.in: an example of an input file with a scoring matrix.
    - RR4.in: an example of an input file with background frequencies.
    - alphabet_ACGT.in: an example of an input file with alphabet letters.
    - seq.in: an example of an input file with a sequence in FASTA format.
    - mode_A.bat, mode_B.bat, mode_C.bat: Windows batch files to run examples in modes A, B, C respectively.

- No special installation is required.
- The executable files can be downloaded, unzipped and run with the appropriate command line.

- Alternatively, the source C++ files can be downloaded, unzipped, and complied in a suitable C++ environment.

Remark.

To compile the C++ files under UNIX, please replace the line

#define _MSDOS_

by the line

//#define _MSDOS_

in the file “sls_auxiliary_repwords.h”.